

Python Basic Commands Cheat Sheet

Numbers & Strings

```
x=10
x+2 ; x-2 addition & Subtraction
x*y Multiplication
x**3 Exponentiation
x%2 Remainder
x/2 Division
x/float(2) Float division
name='Sheldon' Defining
name[0] Accessing
msg[0:9] Substring
len(msg) Length
'Sheldon' + ' Cooper' Concatenation
```

Lists and Tuples

```
mylist1=['Sheldon','Male', 25]
mylist1[0] Accessing elements
mylist1 [0:2] Sub list
mylist1 [2]=35 Updating
len(mylist1) Length of list
del mylist1[5] Deleting
final list = mylist1 + ['L.A','No 173',
"CR108877"]
my_tuple=('Mark','Male', 55)
my_tuple[1] Accessing
len(my_tuple) Length
Tuples can't be updated
```

Dictionaries

```
my_dict={'1':"David", 6:"Bill", 9:"Jim"}
Defining dictionary
my_dict [6] call by the key, not index
my_dict [9] = bob Updating values
my_dict [2] = Kenny Adding key-values
my_dict.keys()
my_dict.values()
```

Conditions and Loops

If Else Condition

```
if age<50:
    print("Group1")
else:
    print("Group2")
```

For Loop

```
for i in range(1,20):
    my_num=my_num+1
    print("my num value is", my_num)
    For loop on a list
```

```
a_list = []
for i in range(1,5):
    a_list.append(i*2)
One liner code for the same
a_list = [i*2 for i in a_list]
```

Function

Defining

```
def remainder(var1, var2):
    a = var1%var2
    return a
```

Applying a function on dataframe
df.apply(np.mean) Returns mean of each column
df.apply(np.mean, axis=1) Returns mean of each row

Packages

```
pip install pandas Installing a new package
import pandas Importing package
import pandas as pd Giving short name to package:
from sklearn.linear_model import LinearRegression Importing a class from a package
from math import * Importing all functions from packages in base form
```

Data Importing and Exporting

```
import pandas as pd
Creating dataframe form dictionary
data = {'name': ['Stan', 'Kyle', 'Eric', 'Kenny'],'age':[9, 9, 11, 12]}
df = pd.DataFrame(data)
```

Importing Data

```
df = pd.read_csv("file.csv") From csv file
df = pd.read_excel("file.xlsx", "sheet name") From an Excel file
```

Exporting Data

```
df.to_csv(file.csv) To a CSV file
df.to_excel(file.xlsx) To an Excel file
```

Checklist and Summary

```
df.shape Dimention of dataframe
df.columns.values columns names
df.head() First few observations
df.tail() Last few Observation
df.dtypes Data Types of variables
df.describe() Summary of all variables
```

```
columns(numerical)
df['column'].describe() Summary of single column
df.column.value_counts() frequency table of a variable
sum(df.column.isnull()) Counting missing values in a column
```

statinfer.com

Training and R&D



Selecting and Sub Setting

```
df1 = df[['col', 'col2']] By selecting columns
df1 = df[['col1', 'col2']][0:10] By Selecting rows and columns
df1=df.drop(['col1'], axis=1)[0:10] Selected rows and excluding columns
```

Subset with variable filter conditions

```
df1=df[df['col1']>40] Single condition
df1 = df[(df['col1']>40) & (df['col2']=="no") | (df['col3']=="male") ]
```

New column with calculated fields

```
df['volume']=(df['length']*(df['width'])*(df['height']))
```

Sorting and Duplicates

```
df=df.sort('col1') Ascending order
df=df.sort('col1',ascending=False) Descending order
Identifying & Removing Duplicates
sum(df.duplicated()) Identify duplicates
df=df.drop_duplicates() Removing duplicates
dupe_id=df.col1.duplicated() Identify duplicates based on a key column
df_uniq=df.drop_duplicates(['col1']) Drop duplicates based on a column
```

Appending, Joining and Merging

```
df1.append(df2) Appending new rows (columns must be same)
df1.join(df2, on='col1', how='inner') Using join function ('left', 'right', 'outer', 'inner')
df_merge = pd.merge(df1,df2 on='col1', how='inner') Using merge function ('left', 'right', 'outer', 'inner')
```